12 Feb 2021 Stable Matching Analysis

Announcements:

1. Problem Set 1 due Thurs 2/18 11:59 pm. (on CMS)

2. Partner finding form:
https://forms.gle/c3AX4UQgTw6FJ3568
(see Ed Discussions post)

3. Email me (rdk2) and Corey Torres (ct635) if you aren't yet on CMS.

Recap of stable matching.

- n firms $f_1, \ldots, f_n$
- n workers $w_1, \ldots, w_n$
- input data specifies a complete ranking of workers for each firm, and a complete ranking of firms for each worker.
- output is a set of (f_i, w_j) pairs expected to satisfy

  (a) PERFECT MATCHING.

  Each party (firm or worker) belongs to exactly one pair.

  (b) STABLE

  There is no "blocking pair":
  if (f, w) and (f', w') are two distinct pairs in the output then it is not the case that

  f prefers w' to w

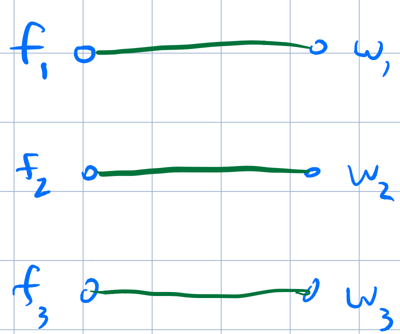
  and

  w' prefer f to f'

For every unmatched pair, one party prefers their match to the other party.

	1st	2nd	3rd
f_1	w_2	w_1	w_3
f_2	w_3	w_2	w_1
f_3	w_3	w_1	w_2

	1st	2nd	3rd
w_1	f_3	f_2	f_1
w_2	f_2	f_3	f_1
w_3	f_1	f_3	f_2



This is an example of running the Gale–Shapley "Proposal Algorithm."

Last time: this alg. always terminates after $O(n^2)$ steps.

Still need to show: the output is a perfect matching, and it is stable.

Obs. 1. The first time a worker receives an offer, it becomes matched. After that, it remains matched, and the ranking of its "mate" never gets worse.

Induction Hypoth: After k loop iterations, any worker who has received an offer is matched to the highest ranked firm that yet made them an offer.

Obs. 2. The set of matched pairs at the end of every loop iteration is a matching: no firm or worker belongs to more than one pair.

Obs. 3. When the algorithm terminates it outputs a perfect matching.

PROOF. Given obs 2, we only need to worry about showing that every party is matched at the end.

\# firms = # workers, so it suffices

to show all firms are matched.
Proof by contradiction: suppose f is not matched. By the termination condition, every firm is either matched or has made an offer to every worker. So f made offers to every worker.
Obs 1 $\Rightarrow$ every worker is matched.
firms = # workers $\Rightarrow$ every firm is matched. (Contradicts assumption f is not matched.)
Consequence: every f is matched at the end of the algo.

Obs 4. The matching that the algorithm outputs has no blocking pairs.

Proof. Suppose that (f, w') is any unmatched pair with
f matched to w and
f' matched to w'
in the algorithm's output. If f prefers w' to w $\Rightarrow$ f made an offer to w' before w.
Obs. 1 $\Rightarrow$ w' is always matched to

the highest ranked firm that made them an offer.

$\implies$ w' prefers f' to f.

$\implies$ (f, w') is <u>not</u> a blocking pair.

[The proof shows for all pairs (f, w') that aren't matched either f prefers its match ~~to~~ w' or w' prefers its match to f.]

One more interesting thing...

<u>Def.</u> Say w is a <u>stable partner</u> for f if there exists a stable perfect matching containing (f, w).

<u>Obs. 5.</u> Running the Gale-Shapley algorithm results in pairing each f with their <u>highest ranked</u> stable partner and pairing each w with their <u>lowest ranked</u> stable partner.

(See book for proof.)

Obs. 6. The output of the G.-S. algorithm doesn't depend on the order in which offers are made.

Proof. Obs. 5 characterizes the output matching using a property that doesn't depend on offer order.